PATENT   Atty Docket No.: 10008025-1
                                            App. Scr. No.: 09/964,769

REMARKS

Favorable reconsideration of this application is respectfully requested in view of the

claim amendments and following remarks. Claims 1-28 are pending in the present

application of which claims 1 and 13 are independent. Claims 1-5 and 13-17 have been

amended. No new matter has been added.

Applicants note with appreciation the Examiner's withdrawal of the previous

objections to the specification and claims, as well as withdrawal of the rejection of claims 1-

12 under 35 U.S.C. § 101.

Claims 1-3, 6-9, 12-15, 18-21, 24 and 26-28 stand rejected under 35 U.S.C. § 102(e)

as allegedly being anticipated by Loginov (U.S. Patent No. 6,567,831) ("Loginov"). Claims

4, 5, 10, 11, 16, 17, 22 and 23 stand rejected under 35 U.S.C. § 103(a) as allegedly being

unpatentable over Loginov in view of Aho et al. ("Compilers: Principles, Techniques, and

Tools", ISBN 0-201-10088-6) ("Aho"). These rejections are respectfully traversed for at

least the following reasons.

Claim Rejection under 35 U.S.C. 102

The test for determining if a reference anticipates a claim, for purposes of a rejection

under 35 U.S.C. § 102, is whether the reference discloses all the elements of the claimed

combination, or the mechanical equivalents thereof functioning in substantially the same way

to produce substantially the same results. As noted by the Court of Appeals for the Federal

Circuit in *Lindemann Maschinenfabrick GmbH v. American Hoist and Derrick Co.*, 221

USPQ 481, 485 (Fed. Cir. 1984), in evaluating the sufficiency of an anticipation rejection

under 35 U.S.C. § 102, the Court stated:

7

PATENT   Atty Docket No.: 10008025-1
                                        App. Ser. No.: 09/964,769

> Anticipation requires the presence in a single prior art reference
> disclosure of each and every element of the claimed invention,
> arranged as in the claim.

Therefore, if the cited reference does not disclose each and every element of the
claimed invention, then the cited reference fails to anticipate the claimed invention and, thus,
the claimed invention is distinguishable over the cited reference.

The Office Action sets forth a rejection of claims 1-3, 6-9, 12-15, 18-21, 24 and 26-28
under 35 USC 102(e) as being allegedly anticipated by U.S. Patent No. 6,567,831 to Loginov
("Loginov"). This rejection is respectfully traversed.

Loginov teaches a method for compiling function evaluations performed by a Very
Long Instruction Word (VLIW) processor using parallel computing algorithms (Abstract;
Column 2, lines 38-58). According to Loginov, a VLIW processor uses parallel algorithms to
minimize computation times (Col. 1, lines 24-38). Loginov teaches that, by moving
scheduling tasks to a compiler, in which the compiler performs most instruction scheduling at
compile time rather than execution units performing scheduling tasks at runtime, a VLIW
processor avoids complex circuitry associated with on-chip instruction scheduling logic (Col.
1, lines 34-43). Based on the instruction scheduling performed by the compiler, the
processing performed by the VLIW processor architectures is improved.

Loginov also teaches that the compiler performs function evaluations. In Loginov, a
function, such as sqrt(x), is evaluated by representing the function as a series expansion, such
as shown in column 3, lines 15-20. Values for one part of the series expansion of equation 1
are computed and stored in a table. The remaining part of the series expansion is a
polynomial, as shown in lines 25-30. The polynomial can be quickly computed using parallel
computation schemes, which greatly reduces processing time. (Col. 3, lines 45-55).

8

PATENT

In contrast, according to an embodiment described in the Applicants' specification, a compiler is provided to optimize the compiling of a family of related functions. A member recognizer is configured to recognize a member function from the family of related functions and a family start caller is configured to make a family-start function call for the family of functions related to the member function. For example, the following statements may appear in a computer program:

X = sin(theta);

Y = cos(theta);

The statement X = sin(theta) would be recognized as being a member of a known family of related functions, namely the trigonometric family of functions. This step of recognizing the statement X = sin(theta) as a member function of the trigonometric family of functions is performed by a member recognizer configured to recognize a member function from the family of related functions. A family start caller, configured to make a family-start function call for the family of functions related to the member function, then makes a family-start function call for the trigonometric family of functions related to the recognized member function, specifically the member function recognized by the statement X = sin(theta). A member finish caller then makes a member-finish function call for the recognized member function, i.e. the statement X = sin(theta). The compiler thus recognizes the member function and performs the appropriate family-start and member-finish function calls to give the following result for the X = sin(theta) member function:

R1 = call_trigstart(theta);

R2 = call_sinfinish(R1);

9

PATENT

The call instruction R1 = call_trigstart(theta) is the appropriate family-start function call for the trigonometric family of functions, and the call instruction R2 = call_sinfinish(R1) is the appropriate member-finish function call for the X = sin(theta) member function. The program statement Y = cos(theta) would also be recognized, by a member recognizer configured to recognize a member function from the family of related functions, as being a member of the same trigonometric family of functions. A family start caller, configured to make a family-start function call for the family of functions related to the member function, then makes a family-start function call for the trigonometric family of functions related to the recognized member function Y = cos(theta). A member finish caller then makes a member-finish function call for the recognized member function Y = cos(theta). The compiler thus recognizes the member function and performs the appropriate family-start and member-finish function calls to give the following result for the Y = cos(theta) member function:

R3 = call_trigstart(theta);

R4 = call_cosfinish(R3);

The call instruction R3 = call_trigstart(theta) is the appropriate family-start function call for the trigonometric family of functions, and the call instruction R4 = call_cosfinish(R3) is the appropriate member-finish function call for the Y = cos(theta) member function.

Prior to optimizing the function calls by the compiler, the program statements are:

R1 = call_trigstart(theta);

R2 = call_sinfinish(R1);

R3 = call_trigstart(theta);

R4 = call_cosfinish(R3);

PATENT

Thus, the _trigstart(theta) call is performed twice. According to an embodiment, all instructions, including the family-start calls call_trigstart(theta), are subject to optimization techniques, including for example elimination of code that is redundant, by common subexpression elimination, code motion, and dead-code elimination techniques. An elimination routine would recognize that R1 and R3 are identical, and would eliminate the repetitive code, transforming the above code to look like:

R1 = call_trigstart(theta);

R2 = call_sinfinish(R1);

R4 = call_cosfinish(R3);

Thus, the compiler operates to reduce the total number of instructions to be performed. For example, R1 and R3 are identical instructions and R3 appeared 48 times in the code. Eliminating R3 by implementing an optimizing technique translates to elimination of 48 instructions that prior to elimination needed to be completed by the computer architecture. This elimination of redundant or repetitive instructions by the compiler, as illustrated by the example discussed above, enhances the processing speed by the computer architecture using the compiler, and also reduces the time and resources necessary for generating an executable file. Furthermore, any combination of a member recognizer, a family start caller, and a member finish caller may be incorporated into a front end of the compiler. Although the above example discusses the trigonometric family of functions, the compiler may compile any family of related functions.

Claim 1 has been amended to further recite "an optimizer configured to optimize said family-start function call; wherein the optimized family-start function call causes execution of instructions that are common to the family of related functions to occur prior to execution

11

PATENT

of instructions for each of a plurality of member-finish functions to reduce a number of instructions executed by the computer architecture in computing more than one member function from said family of related functions.

Similarly, claim 13 has been amended to recite, "optimizing said family-start call to cause execution of instructions that are common to the family of related functions to occur prior to execution of instructions for each of a plurality of member-finish functions to reduce a number of instructions executed by a computer architecture in computing more than one member function from sail family of related function. Support for these amendments is found at least at page 7, lines 5-13 of Applicant's disclosure.

Applicant respectfully submits that Loginov fails to teach an optimized family-start function call causes the execution of the instructions that are common to the family of related functions to occur prior to the execution of instructions for each of a plurality of member-finish functions to reduce the number of instructions executed by the computer architecture in computing more than one member function. Instead, Loginov teaches individually evaluating a function, such as sqrt(x), by expanding the function into a series expansion and performing parallel processing on a portion of the series expansion to improve processing time in a parallel processing computer architecture.

Furthermore, the rejection cites column 2, lines 52-55 and paraphrases this passage stating "overall improvement in processing speed in the evaluation of certain (families of) functions is achieved by (recognizing a member function from a family of related functions, then) representing each function as a series expansion." It appears the rejection is alleging that a function disclosed in Loginov, such as sqrt(x), is the claimed family of related

12

PATENT   Atty Docket No.: 10008025-1
                                                                                            App. Ser. No.: 09/964,769

functions, and the series expansion disclosed by Loginov includes recognizing a member
function from a family of related functions.

However, Loginov merely discloses and describes sqrt(x) and other functions as a
single function, not as a family of related functions. Furthermore, the series expansion
disclosed by Loginov does not recognize a member function from a family of related
functions. Instead, the series function is simply another representation of the function, such
as shown in equation 1 of Loginov. Thus, Loginov fails to teach a member recognizer
configured to recognize a member function from a family of related functions. Loginov also
fails to teach a family of related functions or recognizing a specific function from the family.
Loginov does not disclose that sqrt(x), cbrt(x), and ln(x) are part of a family of related
functions or only applying the function evaluation for these functions.

Loginov also fails to teach "a family start caller configured to make a family-start
function call for said family of related functions" and "a member finish caller to make a
member-finish function call for said member function." Loginov teaches that parallel
algorithms are used to compute "expansion series," wherein each function is represented "as
a series expansion around one or more function argument values." (Column 2, lines 50-58).
In the passage corresponding to column 2, line 66 through column 3, line 1, as cited by the
Examiner, Loginov teaches "the first step of the method…is to divide the range of argument
values for the approximation into n intervals." However, Loginov fails to teach that either an
expansion series or a range of argument values is a "family of related functions," as recited in
claim 1. Loginov also fails to teach that dividing a range of argument values for an
approximation into "n" intervals is identical to making a family-start function call.

13

Furthermore, Loginov fails to teach that either the parallel algorithms, the compiler

itself, the table approximation, the VLIW processor, or any other component of the system

taught by Loginov includes "a member finish caller to make a member-finish function call for

said member function," as instantly claimed. Nowhere does Loginov teach that a compiler

makes a member-finish function call. The compiler taught by Loginov performs function

evaluations, but does not make a member-finish function call or any call for that matter. For

at least the same reasons, Loginov also fails to teach "making a member-finish call for said

member function," as recited in claim 13.

Because claims 2-3, 6-9, 12, and 25-26 incorporate all the limitations of amended

claim 1, and because claims 14-15, 18-21, 24, and 27-28 incorporate all the limitations of

amended claim 13, Loginov fails to teach the invention claimed in claims 1-3, 6-9, 12-15, 18-

21 and 24-28 for at least the reasons given above. Therefore, Loginov does not anticipate the

subject matter of claims 1-3, 6-9, 12-15, 18-21 and 24-28 as amended. Claims 1-3, 6-9, 12-

15, 18-21 and 24-28 are thus allowable over Loginov, and withdrawal of the rejection is

respectfully requested.

Claim Rejection Under 35 U.S.C. §103

The test for determining if a claim is rendered obvious by one or more references for

purposes of a rejection under 35 U.S.C. § 103 is set forth in MPEP § 706.02(j):

> To establish a _prima facie_ case of obviousness, three basic criteria
> must be met. First, there must be some suggestion or motivation,
> either in the references themselves or in the knowledge generally
> available to one of ordinary skill in the art, to modify the reference
> or to combine reference teachings. Second, there must be a
> reasonable expectation of success. Finally, the prior art reference
> (or references when combined) must teach or suggest all the claim
> limitations. The teaching or suggestion to make the claimed
> combination and the reasonable expectation of success must both

PATENT   Atty Docket No.: 10008025-1
                                                App. Ser. No.: 09/964,769

be found in the prior art and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Therefore, if the above-identified criteria are not met, then the cited reference(s) fails to render obvious the claimed invention and, thus, the claimed invention is distinguishable over the cited reference(s).

The Office Action sets forth a rejection of claims 4, 5, 10, 11, 16, 17, 22 and 23 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Loginov in view of Aho. This rejection is respectfully traversed.

As discussed above, claim 1 has been amended to further recite "an optimizer configured to optimize said family-start function call; wherein the optimized family-start function call causes execution of instructions that are common to the family of related functions to occur prior to execution of instructions for each of a plurality of member-finish functions to reduce a number of instructions executed by the computer architecture in computing more than one member function from said family of related functions.

Similarly, claim 13 has been amended to recite, "optimizing said family-start call to cause execution of instructions that are common to the family of related functions to occur prior to execution of instructions for each of a plurality of member-finish functions to reduce a number of instructions executed by a computer architecture in computing more than one member function from sail family of related function. Support for these amendments is found at least at page 7, lines 5-13 of Applicant's disclosure.

Neither Aho nor Loginov teach or suggest the features of independent claims 1 and 13 as amended, and thus dependent claims 4, 5, 10, 11, 16, 17, 22 and 23 are believed to be allowable.

15

PATENT

Furthermore, it would not have been obvious to combine Aho with Loginov. Loginov is directed to the use of parallel algorithms to compute "expansion series," wherein each function is represented "as a series expansion around one or more function argument values." (Column 2, lines 50-58.) The series expansion is performed in parallel. (Column 3, lines 51-54). In contrast, Aho is directed to compiler optimization techniques for serial execution. For example, Aho teaches an "in-line" expansion technique that adds code to a procedure body instead of making a function call but which reduces the execution time of a program. (Aho, p.428, lines 24-34.) Aho further teaches that the efficiency gained by in-line (i.e., serial) expansion is due to eliminating the "cost associated with setting up an activation procedure – space is allocated for the activation record, the machine status is saved, links are set up, and then control is transferred." (Aho, p.428, lines 36-38.) One skilled in the art would have recognized that similar set-up costs are required for parallel processing execution. Thus, one skilled in the art would not have been motivated to combine Loginov and Aho because Aho teaches away from the parallel execution described by Loginov.

Furthermore, with regard to claim 4, it bears repeating that Aho was cited in the rejection to teach that an optimizer is configured to in-line expand at least one of said family-start or member-finish calls. However, Aho discloses in-line expansion may be used for reducing the running time of a program (page 428, Aho), in the context of in-line expansion in call-by-name procedures (pages 428) in a run-time environment. However, Loginov discloses function evaluation performed in a compile-time environment and not a run-time environment. Thus, it would not have been obvious, if not impossible, to perform the in-line expansion described in Aho in the compile-time environment for the VLIW processor of Loginov.

PATENT

With regard to claim 5, the Examiner paraphrases page 592, lines 10-12 of Aho, and

states that Aho teaches subexpression elimination, code motion, and dead-code elimination as

common examples of function preserving transformations. However, Aho fails to teach or

suggest that subexpression elimination, code motion, and dead-code elimination may be used

as function preserving transformations specifically by a compiler <u>configured to optimize said

family-start call</u>, as recited in the claims.

With regard to claim 10, the Examiner paraphrases page 463, lines 1-3 of Aho, and

states that Aho teaches making function calls in an intermediate language. However, Aho

fails to teach a compiler used by a computer architecture to compile a family of related

functions, comprising "a member recognizer configured to recognize a member function from

said family of related functions," "a family start caller configured to make a family-start

function call for said family of related functions" and "a member finish caller to make a

member-finish function call for said member function," wherein one or both of said family

start caller and said member finish caller are configured to make said family-start and

member-finish function calls, respectively, in an intermediate language. Furthermore, with

regard to claim 11, Aho also fails to teach or suggest that said intermediate language is non-

architecture specific and non-operating system specific. Although the Examiner paraphrases

page 463, lines 1-3 of Aho, nowhere does Aho specifically teach that the front end of a

compiler translates a source program into *an non-architecture specific and non-operating

system specific* intermediate language representation from which the back end generates

target code, as recited in claim 11. Since Aho also fails to remedy the deficiencies of

Loginov, the Applicants respectfully request withdrawal of the rejection.

PATENT

With regard to claims 16, 17, 22 and 23, the Examiner only alleges that "the Loginov/Aho combination also discloses such claimed limitations as addressed in claims 4, 5, 10 and 11, respectively." However, the Examiner provides no specific teaching or suggestion by Loginov or Aho, whether alone or in combination, to support the rejection. For at least the reasons discussed above, with regard to claims 4, 5, 10 and 11, respectively, the Applicants respectfully submit that neither Loginov or Aho, whether alone or in combination, teach or suggest the subject matter of claims 16, 17, 22 and 23.

Furthermore, Aho was cited merely to provide definitions of certain terms, e.g. call-by-name procedures (pages 428), function preserving transformations (page 592) and intermediate code generation (page 463). However, Aho describes these terms in the context of a run-time environment, and not in the context of a compiler. Because the instant claims pertain to a compiler used by a computer architecture to compile a family of related functions, and not to a run-time environment, the Applicants submit that Aho fails to provide any teaching or suggestion to render the claims obvious and it would not have been obvious to combine Aho with Loginov.

Therefore, because neither Aho nor Loginov teach or suggest the features of independent claims 1 and 13 as amended, dependent claims 4, 5, 10, 11, 16, 17, 22 and 23 are believed to be allowable over Loginov in view of Aho for at least this reason, as well as for the additional reasons discussed above.

18

PATENT   Atty Docket No.: 10008025-1
                                                    App. Ser. No.: 09/964,769

Conclusion

In light of the foregoing, withdrawal of the rejections of record and allowance of this

application are earnestly solicited. Should the Examiner believe that a telephone conference

with the undersigned would assist in resolving any issues pertaining to the allowability of the

above-identified application, please contact the undersigned at the telephone number listed

below. Please grant any required extensions of time and charge any fees due in connection

with this request to deposit account no. 08-2025.

                                        Respectfully submitted,

Dated: June 23, 2005   By   _____
                                        Ashok K. Mannava
                                        Registration No. 45,301

                                        MANNAVA & KANG, P.C.
                                        8221 Old Courthouse Road
                                        Suite 104
                                        Vienna, VA 22182
                                        (703) 652-3822
                                        (703) 880-5270 (facsimile)

19